Combining nonlinear multiresolution system and vector quantization for still image compression

Yiu-fai Wong
Lawrence Livermore National Laboratory
Institute for Scientific Computing Research
Livermore, CA 94550

ABSTRACT

It is popular to use multiresolution systems for image coding and compression. However, general-purpose techniques such as filter banks and wavelets are linear. While these systems are rigorous, nonlinear features in the signals cannot be utilized in a single entity for compression. Linear filters are known to blur the edges. Thus, the low-resolution images are typically blurred, carrying little information. We propose and demonstrate that edge-preserving filters such as median filters can be used in generating a multiresolution system using the Laplacian pyramid. The signals in the detail images are small and localized to the edge areas. Principal component vector quantization (PCVQ) is used to encode the detail images. PCVQ is a tree-structured VQ which allows fast codebook design and encoding/decoding. In encoding, the quantization error at each level is fed back through the pyramid to the previous level so that ultimately all the error is confined to the first level. With simple coding methods, we demonstrate that images with PSNR 33 dB can be obtained at 0.66 bpp without the use of entropy coding. When the rate is decreased to 0.25 bpp, the PSNR of 30 dB can still be achieved. Combined with an earlier result, our work demonstrate that nonlinear filters can be used for multiresolution systems and image coding.

1. INTRODUCTION

One generic approach for image compression is to decompose an image into components which hopefully can be encoded efficiently. Multiresolution systems such as subband coding and wavelets (see [1, 2]) allow one to take advantage of the signal characteristics in different frequency bands to achieve compression [3, 4]. However, these techniques are linear. Nonlinear features in the signals cannot be captured and utilized in a single entity for compression. For example, an edge is arguably the most important feature and is nonlinear. Thus, low resolution images are typically blurred in these multiresolution schemes and carry little information. Furthermore, distortion of edges in lossy schemes can have serious consequences on later processing tasks in pattern recognition and machine vision. Thus, it is not clear whether linear techniques are most efficient in encoding nonlinear features.

Although it is common to equate frequency bands with various resolutions, it is doubtful if this resembles the mechanism of human vision at all. For example, when a person looks at a mountain far way, one definitely sees the sharp boundary between the mountain and the sky. One does not see a blurred mountain top. Thus, an image at a coarse scale or low resolution, though blurred and missing the fine details, still has well-defined boundaries and edges. This points to the necessity of edge-preserving filtering as resolution decreases, which forms the motivation for our work here.

The idea here then is to use edge-preserving filters in generating a multiresolution system such that edges are preserved in the low-resolution description. This has been demonstrated previously using the clustering filter [5]. Furthermore, coding schemes for the subimages obtained by the nonlinear decomposition were attempted in [6]. It was shown that good compression ratio and image quality could be achieved. However, one drawback in [6] is that filtering is not fast enough to meet real-time demands, although decoding is simple and fast. Thus, we seek the use of other fast nonlinear filters. Nonlinear multiresolution systems have previously been investigated [7, 8, 9]. Morphological operations are used in the pyramidal decomposition [7]. In [9], stack filters are designed to satisfy the minimum mean absolute error criterion. Our nonlinear system differs from previous attempts because we consider it from the viewpoint of edge-preserving filtering.

Once an image has been decomposed into subimages, the next task is to encode the signals, for which many

0-8194-1481-6/94/$6.00

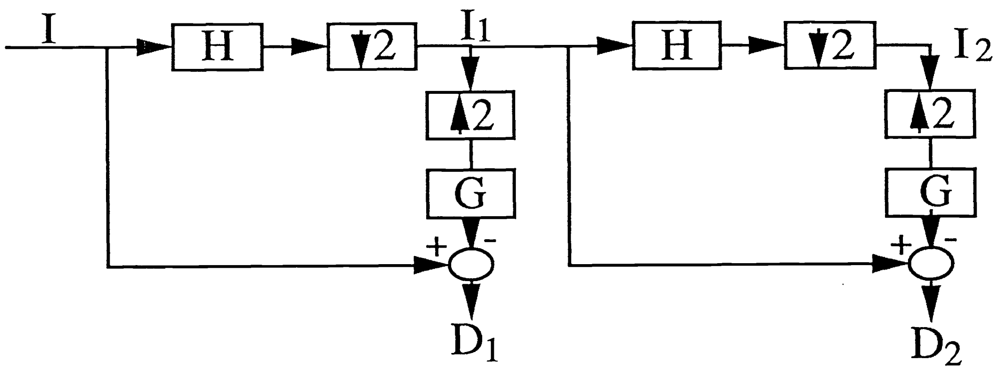

Figure 1. A Laplacian pyramid using the clustering filter.

methods exist [13]. Vector quantization has been used by many researchers for encoding the subimages in various decomposition schemes [10, 11, 12, 4] because it is more advantageous to code vectors than scalars. To reduce the complexity of codebook design and encoding/decoding, a very fast tree-structured scheme called Principal Component Vector Quantization (PCVQ) is used here. This method also allows one to design one codebook for each image, thus making it more adaptive to the changes in the images.

This paper is organized as follows: Section 2 contains a description of the nonlinear multiresolution system and its decomposition of real data. PCVQ will be described in Section 3. In Section 4, we describe the encoding schemes and the results. A main contribution is the demonstration that a simple nonlinear multiresolution systems can be effective for image compression.

2. MEDIAN FILTERS FOR NONLINEAR MULTIRESOLUTION SYSTEM

The class of edge-preserving nonlinear filters is very broad [14]. Since its introduction by Tukey [15], median filters have been widely applied and many variations have been investigated [16, 17, 18, 19]. In median filtering, one slides a window of size N across the whole image. The data points in each window are sorted and the median is used as the output. Since speed is a major concern in image coding for practical reasons, it may seem deceptively simple to suggest the use of median filters. We will show that this is possible.

Given that the filter is nonlinear, it is not clear how to design a maximally decimated system and reconstruct the signal perfectly. It turns out that one of the most popular multiresolution systems, the Laplacian pyramid [20], always allows perfect reconstruction at the expense of oversampling. It has been noted by several previous investigators that the pyramid structure is very flexible in that any filters can be used for either the **reduce** or **expand** operations at each stage of the pyramid [8, 21]. Thus, one can simply use the median filter for the **reduce** step. For the **expand** step, we choose to use a simple bilinear interpolation filter, instead of nonlinear filter because both quantization and its effects can be highly nonlinear. Our version is shown in Figure 1. We find that a 3 by 3 median filer works well for our system.

Briefly, the system works by filtering an original image I by the median filter H to get I_{1h}, which is decimated by a factor of 2 in each dimension to get I_1:

$$I_1 = H(I)_{\downarrow 2} \tag{1}$$

where $\downarrow 2$ denotes decimation. One then expands I_1 to the original size of I by inserting zeros between the samples of I_1. For each location where a zero is inserted, its interpolated output is computed by a bilinear filter G on I_1. G can be thought of convolving $g_1 = \{1/2, 1, 1/2\}$ in both the horizontal and vertical directions. Let I_{0i} denote the interpolated image. The difference signal D_1 is defined as

$$D_1 = I - I_{0i}. \tag{2}$$

For an L-level system, this operation is repeated $L-1$ times to give a set of signals $J = \{D_1, D_2, \ldots, D_{L-1}, I_{L-1}\}$.

One can verify that this architecture indeed guarantees perfect reconstruction.To reconstruct I_k from I_{k+1} and D_{k+1}, we compute the interpolated image I_{ki}. By definition, $I_k = D_{k+1} + I_{ki}$. Furthermore, due to the simplicity of the interpolation filter G, the reconstruction process is extremely fast.

Let us now apply this scheme to real data shown in Figure 2. Figures 3a-c show the set of filtered images I_is of Figure 2. The difference signals D_is (absolute values stretched to 8 bits) are shown in Figure 4. The energy of the signals in Figure 3 are $27.77, 50.96$ and 105.56 respectively. We note that the signals in the difference images are highly localized.

One may ask how efficient this decomposition is. For this, we compare it with the original Laplacian pyramid [20] which has been shown to be very efficient. With $\alpha = 0.6$, the energies of the difference signals are 22.35, 44.54 and 86.71 respectively. Let us compute the first-order entropy of the subimages. The entropies of the Laplacian images are $3.763(D_1), 4.055(D_2), 4.677(D_3), 7.124(I_3)$ bits respectively, while those obtained by the nonlinear multiresolution subimages are 3.766, 3.725, 4.224, 7.079 bits respectively. The overall average entropy/pixel for the two systems are 5.180 and 5.072 bits respectively. Thus, there is a saving of 0.108 bits/pixel by using nonlinear filters. This shows that to a first-order approximation, the decomposition of the nonlinear multiresolution system may be more efficient than the Laplacian pyramid.

3. PRINCIPAL COMPONENT VECTOR QUANTIZATION

Vector quantization (VQ) has been proposed for image compression and many algorithms have been invented (see [22, 23] for comprehensive reviews). A VQ algorithm can generically be described as follows: Given a set S of feature vectors extracted from one or more images, compute M codewords $C = \{y_1, \ldots, y_M\}$ such that the total distortion over the training set is minimized given a distortion measure $d(x, y)$ of encoding x by y. Obvious, the complexity of codebook design has to do with the size $|S|$ of the training set S, the dimension d of a feature vector, the codebook size M and the structure of the codebook. The complexity of encoding is dependent on the encoding rule, M and d. A common choice of distortion measure is the squared Euclidean distance $d(x, y) = ||x - y||^2$.

Among the numerous algorithms, the LBG (Linde-Buzo-Gray) algorithm [24], also known as Generalized Lloyd Algorithm (GLA), has been most popular for designing codebooks. Although GLA finds reasonable local minima, its speed is quite slow, placing a severe constraint on codebook size M and vector dimension d. Another undesirable feature of GLA is that the codebook is unstructured. A time consuming full search is needed for encoding when M and d are large. Since speed is a major concern in real-time signal transmission, fast algorithms which find good codebooks and allow fast encoding are highly desirable.

Tree-structured VQ (TSVQ) is one example of structurally constrained VQ which alleviates both codebook design and encoding. Several researchers have proposed TSVQ by splitting a codeword along the principle components of the set of vectors it encodes [28, 26, 27, 29, 30, 31]. In [26, 29], the algorithm complexity was identified as $O(d^2|S|\log(M))$, while noting that the computation for finding the principal components is negligible in the overall complexity. This is important because it shows that PCVQ is much faster than GLA because a single pass of GLA would require $\mathcal{O}(d|S|M)$ operations. However, Wu and Zhang [26] used GLA for further optimization of the codewords after the splitting along the principal components, thus slowing down the method. Their implementation is 50% slower than the PNN algorithm [32]. Huang and Huang [27] used neural network learning algorithm to find the principal components, which is certainly slow. Orchard and Bouman [29] were primarily concerned with encoding color images and thus they used vectors of dimension 3 only. It is clear that the power of PCVQ has not been fully explored.

In this work, we use a straightforward and fast implementation of PCVQ and demonstrate that PCVQ is capable of finding good codebook much faster than any other algorithms. As PCVQ has been more or less described in [26, 27, 29], we only give a brief description here.

For a given set of data S, its covariance matrix is given by

$$C = \frac{1}{|S|}\sum_i (x_i - \bar{x})(x_i - \bar{x})^t \tag{3}$$

where $\bar{x} = \frac{1}{|S|}\sum_i x_i$. The eigenvector u corresponding to the largest eigenvalue of C is called the *principal component* of S. A physical interpretation of principal component is that the projection of the data S onto a one-dimensional subspace has the largest variance if this subspace coincides with u.

PCVQ divides the data S into two subsets S_2 and S_3 by the hyperplane perpendicular to the principal component while passing through the centroid of the data [29]. Thus,

$$\begin{aligned} S_2 &= \{x \in S : (x - \bar{x})^t u < 0\} \\ S_3 &= \{x \in S : (x - \bar{x})^t u \geq 0\}. \end{aligned} \tag{4}$$

The total distortion for a subset S_i is defined as

$$D(S_i) = \sum_{j \in S_i} ||x_j - \bar{x}||^2. \tag{5}$$

The algorithm proceeds by selecting the leaf having the largest distortion and split it along its principal component. At any stage, the leafs contain the codewords and the subsets they encode. There are alternatives for choosing the best leaf to split. In [27, 29], the leaf with the largest eigenvalue is chosen. One can also choose the leaf that results in largest reduction in distortion after splitting. Our method is the simplest one.

Given a training set S and codebook size M, our implementation of PCVQ is as follows:

1. Let the initial leaf be $S_1 = S$;
2. Do the following $M - 1$ times:
   (a) Find the n leaf such that its distortion is maximum among all the leafs;
   (b) Split S_n into S_{2n} and S_{2n+1};
3. Stop.

In computing the covariance matrices, one can use C_n and C_{2n} to determine C_{2n+1}. The principal component is computed by power method [33] which is very fast.

One can see the main difference between our implementation of PCVQ and those in [26, 27]. The major motivation is to split the subsets along the principal components and take advantage of the speed of the power method. We are not concerned with optimization here. After all, although the GLA sets out to optimize the total distortion in some sense, its convergence to local minima means that it is not clear at all what GLA is really minimizing. In testing PCVQ using blocks extracted from real images, we found that its performance is about 0.5 dB worse than those obtained by GLA. Recall that the time complexity of PCVQ is $O(d^2|S|\log(M))$. Thus, PCVQ scales very well with both d and M.

4. IMPLEMENTATION OF OVERALL CODING SCHEME

Traditionally, codebooks used in VQ are pre-computed from a training set and shared by both the encoder and the decoder. One obvious drawback is that if the distribution of the images changes, a codebook will no longer represent the inputs well unless it adapts. Since the operations of PCVQ are very simple, consisting of mainly matrix multiplications, potentially it can be implemented in hardware. Thus, we propose to design one codebook per image.

To apply PCVQ to the subimages generated by our nonlinear multiresolution system, we note that each block of vector contains difference signals. Since a bilinear filter is used in the interpolation step, most of the energies of the difference signals are concentrated around the edges. Thus, a block near an edge will have a small variance and vice versa. Thus, it may be advantageous to divide the codebook design into two subcodebooks: one for smooth blocks and another for rough blocks. This is in the same spirit as classified VQ (see [23]). One can imagine that a more sophisticated scheme can be used here. However, our emphasis is on the nonlinear multiresolution system. All the operations are kept simple. We do not use entropy or arithmetic coding for compaction, which will result in further reduction of the bit rate.

For $L = 3$, our nonlinear system generates images $J = \{D_1, D_2, I_2\}$. The size of the low-resolution image I_2 is only 1/16 of the original one. Definitely one can use DPCM or adaptive DPCM to code I_2. However, a simple PCM is used here and I_2 is not compressed. To code the difference images, 4×4 blocks of vectors are extracted from the difference images. Using a threshold of 64 on the energy, the vectors are divided into two groups. A codebook is designed for each using PCVQ. Thus, with a combined codebook size of M and vector dimension $d = 16$, it takes about $6Md$ bits to represent the codebooks. This is really negligible compared to the image size. In the test images, the sizes of the codebooks are 32 and 224 respectively and $d = 16$, the cost of the codebook is about 0.09 bpp and the overall compression ratio is about a factor of 12 (0.66 bpp). Another feature of our system is that the quantization error at each stage is fed back to the previous stage. This idea has been used in [10, 21], which is another flexible feature associated with Laplacian pyramid.

Figure 5 shows the compressed image of Figure 2 after applying the coding scheme on the subimages, with a codebook sizes of 24 and 232 for the smooth and rough blocks respectively. PSNR is 33.93 dB. Figure 6a and 6b show the original and compressed peppers images. PSNR is 32.24 dB. There are two parameters one can change in our scheme: the threshold and the codebook sizes. It is important to allocate more codewords to the rough blocks because edges are important. There may be better ways to choose the threshold, but we do not explore it here. To illustrate the effect as bits/pixel decreases, we compute the distortion as the codebook size is decreased by a factor of 2, while keeping the relative ratio of the codebook sizes to 7:1 (rough:smooth). Block size of the vectors is 4×4. The results are listed in Table 1. The image corresponding to the case of .253 bpp is shown in Figure 7.

Codebook Size	8	16	32	64	128
PSNR (dB)	30.40	31.15	31.80	32.52	33.24
bits/pixel	.253	.318	.387	.461	.547

Table 1. PSNR of the compressed girl image at various bpp.

5. DISCUSSION

We have demonstrated a nonlinear multiresolution system based on median filtering for image compression. By using edge-preserving filters, low-resolution images with sharp edges can be generated. Moreover, the difference signals are highly localized to the edges. To encode the images, we use principal component vector quantization method to design codebooks for the difference images. With these simple schemes, we demonstrated that we can encode images images with very good visual quality at fairly low bits per pixel. Since PCVQ is tree-structured, decoding is very simple and fast. Moreover, the VQ algorithm consists of mainly matrix multiplications, making it well suited for hardware implementation. Combined with the results of previous work [6], we show that one can use edge-preserving filters for multiresolution system for image coding. Our scheme can modified for progressive transmission, which is naturally embedded in the pyramid structure [20]. In fact, one can design the codebook iteratively and transmit the codewords progressively too. This however remains an ongoing investigation.

6. ACKNOWLEDGMENTS

I thank the late Dr. Edward Posner of Caltech for his constant encouragement for this research. I also acknowledge help from Kar-Ming Cheung and R.M. Gray with the references on PCVQ. I also thank M.C. Lee for discussions. This work was partially supported by the Lawrence Livermore National Laboratory through DOE contract No. W-7405-ENG-48.

References

[1] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, 8, 14-38, 1991.

[2] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Patt. Anal. and Mach. Intell.*, PAMI-11, pp. 674–693, 1989.

[3] J.W. Woods and S.D. O'Neil, "Subband Coding of Images," *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-34, 1278-1288, 1986.

[4] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image Coding Using Wavelet Transform," *IEEE Trans. Image Processing*, 1, 205-220, 1992.

[5] Yiu-fai Wong, "Nonlinear Scale-space Filtering and Multiresolution System," submitted for publication, 1993.

[6] Yiu-fai Wong and Ming-Chieh Lee, "Image Compression by A Nonlinear Multiresolution System," *Proc. IEEE Workshop on Visual Signal Processing and Communications*, 227-230, Australia, September 1993.

[7] R. Haralick, X. Zhuang, C. Lin and J. Lee, "The Digital Morphological Sampling Theorem," *IEEE Trans. ASSP*, 37, 2067-2090, 1989.

[8] D. Anastassiou, "Generalized three-dimensional pyramid coding for HDTV using nonlinear interpolation," *Proc. Picture Coding Symposium*, 1.2-1-1.2-2, Cambridge, Massachusetts, March 1990. 12, 1206-1210, 1990.

[9] G.C. Furski and M.T. Orchard, "Optimal Stack Filters for Pyramidal Decomposition," *Proc. 25th Conf. on Information Sciences and Systems*, 311-316, March 1991.

[10] L. Wang and M. Goldberg, "Progressive image transmission using vector quantization on images in pyramid form," *IEEE Trans. Comm.*, 37, 1339-1349, 1989.

[11] C.I. Podilchuk, N.S. Jayant and P. Noll, "Sparse codebooks for the quantization of non-dominant sub-bands in image coding," *Proc. ICASSP*, 2101-2104, 1990.

[12] T. Senoo and B. Girod, "Vector quantization for entropy coding of image subbands," *IEEE Trans. Image Processing*, 1, 526-532, 1992.

[13] N. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[14] I. Pitas and A.N. Venetsanopuolos, *Nonlinear Digital Filters–Principles and Applications*, Kluwer, Boston, 1990.

[15] J.W. Tukey, *Exploratory Data Analysis*, Reading, Addison-Wesley, MA, 1977.

[16] T.A. Nodes and N.C. Gallagher,Jr., "Median filters: some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Processing*, 30, 739-746, 1982.

[17] A.C. Bovik, T.S. Huang and D.C. Munson, Jr., "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, 31, 1342-1350, 1983.

[18] J. Astola, P. Heinonen and Y. Neuvo, "Linear median hybrid filters," *IEEE Trans. Circuits and Systems*, 36, 1430-1438, 1989.

[19] Y.H. Lee and S.A. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, 33, 672-683, 1985.

[20] P.J. Burt and E.H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, 31, 532-540, 1983.

[21] K. M. Uz, M. Vetterli and D.J. LeGall, "Interpolative multiresolution coding of advanced television with compatible subchannels," *IEEE Trans. Circuits and Systems for Video Technology*, 1, 86-100, 1991.

[22] N.M. Nasrabadi and R.A. King, "Image Coding Using Vector Quantization: A Review," *IEEE Trans. Communications*, 36, 957-971, 1988.

[23] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Boston, 1992.

[24] Y. Linde, A. Buzo and R.M. Gray, "An algorithm for vector quantization, " *IEEE Trans. Communications*, COM-28, 84-95, 1980.

[25] L. Breiman, J.H. Friedman, R.A. Olshen and C.J. Stone, *Classifications and Regression Trees*, Wadsworth, Belmont, California, 1984.

[26] X. Wu and K. Zhang, "A better tree-structured vector quantizer," *Proc. IEEE Data Compression Conf.*, 392-401, 1991.

[27] S.C. Huang and Y.F. Huang, "Principal Component Vector Quantization for Abrupt Scene Changes," *Proc. ISCAS*, pp. 2280-2283, 1992.

[28] A. Lowry, S. Hossain and W. Millar, "Binary Search Trees for Vector Quantization," *ICASSP Proceedings*, p. 51.8.1-51.8.4, 1987.

[29] M. Orchard and C. Bouman, 'Color Quantization of Images," *IEEE Trans. Signal Processing*, 39, 2677-2690, 1991.

[30] X. Wu, "Vector Quantizer Design by Constrained Global Optimization," *Proc. IEEE Data Compression Conf.*, 132-141, 1992.

[31] K.I. Diamantara and S.Y. Kung, "Compressing moving pictures using APEX neural principal component extractor," *Proc. IEEE Workshop on Neural Networks*, September, 1993.

[32] W.H. Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. Acous., Speech, and Signal Process.* 37, 1568-1575, 1989.

[33] Gene H. Golub and Charles F. van Loan, *Matrix Computations*, Johns-Hopkins University Press, 1983.

Figure 2. A Girl image.

a b

Figure 3. Low resolution images.

a b c

Figure 4. Difference Images generated by the nonlinear multiresolution system.

Figure 5. Compressed girl image at .66bpp, PSNR=33.93 dB.

Figure 7. Compressed girl image at .253bpp, PSNR=30.40 dB.

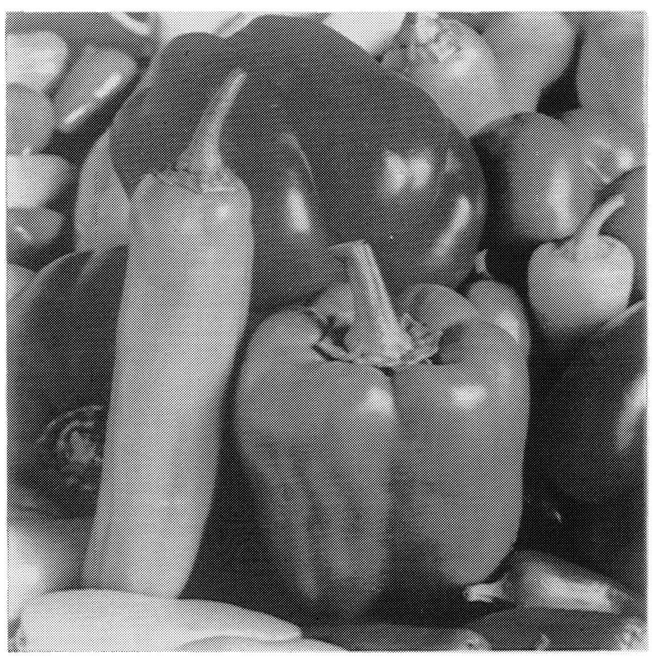

Figure 6a. The original Peppers image.

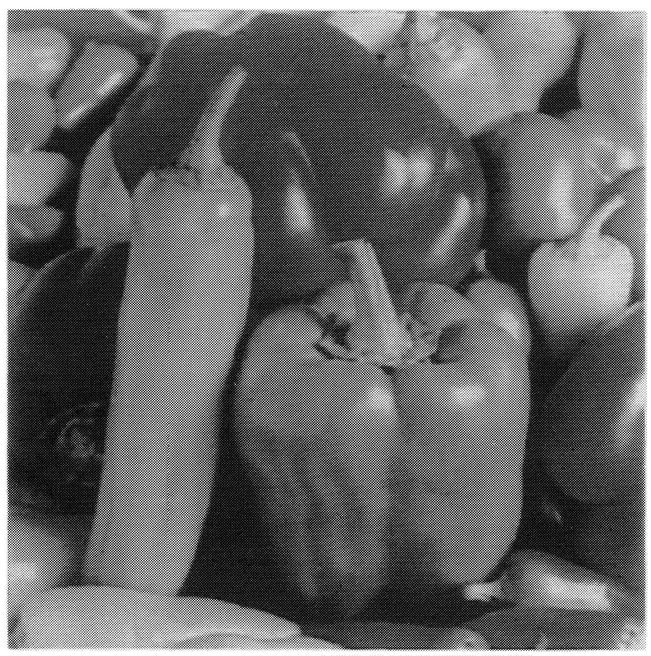

Figure 6b. Compressed Peppers at .66bpp, PSNR=32.24 dB.